Remarks

In the above captioned final Office Action claims 1-56 have been rejected. In response, claims 1, 16, 23, and 29 have been amended and claims 44-55 have been cancelled. Accordingly, claims 1-43 and 56 remain pending in the application.

Claim Amendments

Support for the amendments to claims 1 and 29 may be found in claims 16 and 23.

Claim Rejections under 35 U.S.C. § 103(a)

On page 2 item 3 of the above-identified final Office Action, the Examiner rejects claims 1-56 as being unpatentable over U.S. Patent Publication No. 2003/0200457 to *Auslander et al* (hereinafter "Auslander") in view of U.S. Patent No. 6,578,033 to *Singhal et al* (hereinafter "Singhal") under §103.

Applicants respectfully note that the rejections of claims 44-55 are obviated by their cancellations.

More specifically, in rejecting claim 1, the Examiner cited paragraphs 20 and 22 of Auslander. These passages describe a method for managing access to a lock by a plurality of threads by associating the lock and each thread attempting to access the lock with a pointer structure. The pointer structure of the lock has a head pointer to a first thread in a queue waiting to access the lock and a tail pointer to the last thread in the queue. When the lock is released by a thread, the head pointer of the lock is not reset until the first thread in the queue has acquired the lock. By waiting until the first queued thread has acquired the lock to reset the head pointer, the method of Auslander ensures that the first queued thread, not some other thread happening to try to acquire the lock at the right moment, will in fact acquire the lock. Also, the lock of Auslander has three states (lock not acquired; lock acquired, no one waiting; and lock acquired,

one or more threads waiting) which correspond to various head and tail pointer settings of the lock.

The Examiner, however, admits that Auslander fails to disclose all of the elements of claim 1, and cites Figure 7 and col. 5, line 14 of Singhal as curing the deficiency. That figure and passage disclose a lock having four states. Those states are described in greater detail in col. 10, lines 1-10. They include a first state in which the lock has not been acquired, a second state in which the lock has been acquired, but no threads are waiting, a third state in which the lock has been acquired and one thread is waiting, and a fourth state in which the lock has been acquired and two threads are waiting. The first waiting thread spins at the lock at the highest rate, and the second thread spins at the lock at a second, slightly lower rate. Because order of access is managed by the speed at which the waiting threads spin, no queue is needed. Also, as mentioned on col. 10, lines 8-9, the lock stores a single value representing the state of the lock.

While Applicants continue to disagree with the Examiner's rejection, for reasons discussed below, Applicants nonetheless amended claim 1 to overcome the Examiner's rejection. Claim 1 now recites that "the lock being considered to be in any one of at least four states in any point in time, and each state is represented by a multi-part state value, each multi-part state value including a flag value, a first thread value, and a last thread value." And although Auslander and Singhal do teach a lock which can be in one of four states, they do not teach or suggest each state being represented by a multi-part state value including a flag value, a first thread value, and a last thread value. At best, Auslander and Singhal teach the state being represented by a *single* state value stored by the lock. Nothing in either reference suggests a state value that is multi-part and includes a flag value, a first thread value, and a last thread value.

Further, the use of a flag value, first thread value, and last thread value to represent a lock state was not simply a design choice, but rather an important feature in

implementing the lock as a state machine. The usefulness of this particular multi-part state value in representing the lock states with a minimal amount of data is illustrated in Figure 4 of the instant application, which shows how both the four states of the lock and the waiting queue can be completely described by four simple multi-part state values. Auslander and Singhal, in contrast, disclose only a single state value, as described in Singhal, as representing a lock state, because Auslander and Singhal are not concerned with implementing a state machine which relies on values completely describing both of the lock and its waiting queue. Rather, they are concerned with avoiding allocation of unnecessary data (Auslander) and with managing lock access using thread spinning speeds rather than a queue (Singhal). Thus, one of ordinary skill would not find motivation in either Auslander or Singhal to modify the references to represent a lock state with a multi-part state value including a flag value, a first thread value, and a last thread value.

Additionally, as mentioned above, Applicants disagree with the Examiner's reasons for rejecting claim 1. Specifically, Applicants do not believe that Auslander and Singhal teach or suggest "speculatively determining by the thread, the next state of the lock, where the next state is the state of the lock if the thread proceeds to perform the selected action and the thread is successful", as is claimed by claim 1. In the above-identified rejection, the Examiner cites paragraph 22 of Auslander as teaching the "speculative determining" recited by claim 1.

In fact, as discussed above, paragraph 22 of Auslander discloses a mechanism for secured lock transfer among threads which makes the speculative determining recited by claim 1 unnecessary. That mechanism prevents the head pointer from being reset until the thread pointed to by it has acquired the lock from the releasing thread. Thus, in Auslander, it is not necessary to determine the next state of the lock because only one subsequent state is even possible. In claim 1, in contrast, it is not certain that an acquiring thread, even a thread at the head of the queue, will acquire the lock when it is released. Thus, claim 1 "speculatively" determines the next state (i.e., the state

when the lock is released) before attempting to acquire the lock to increase the chance of success in acquiring the lock.

Further, by providing a mechanism to guarranty successful acquisition of the lock by the first waiting thread, Auslander arguably teaches away from the "speculative determining" of claim 1, which is only performed because acquisition of the lock by the thread of claim 1 is not certain.

Accordingly, claim 1 is patentable over Auslander and Singhal under §103.

Claims 16, 23, and 29 include similar recitations directed to apparatuses and articles of claim 1, Accordingly, for at least the same reasons, claims 16, 23, and 29 are patentable over Auslander and Singhal under §103.

Claims 2-15, 17-22, 24-28, 30-43, and 56 depend from claims 1, 16, 23, and 29, incorporating their recitations. Accordingly, for at least the same reasons, claims 2-15, 17-22, 24-28, 30-43, and 56 are patentable over Auslander and Singhal under §103.

Conclusion

In conclusion, claims 1-43 and 56 are in condition for allowance. Early issuance of Notice of Allowance is respectfully requested.

The Commissioner is hereby authorized to charge shortages or credit overpayments to Deposit Account No. 500393.

Respectfully submitted,

SCHWABE, WILLIAMSON & WYATT, P.C.

Dated: January 12, 2008

/Robert C. Peck/
Robert C. Peck
Registration No. 56,826

Pacwest Center, Suite 1900
1211 SW Fifth Avenue
Portland, Oregon 97204
Telephone: 503-222-9981